

as described herein in Examples 8 and 11, where the storage organization may be multiple disk drives worth of content with several mirrors.

In this latter case, monitoring agent 440 may be used to monitor or keep track of the maximal outstanding I/O for every disk drive and/or sub-disk in the current polling window, for example, using a re-settable outstanding I/O counter or other suitable tracking method. Workload monitor 426 in resource manager 420 may track the total number of viewers and the aggregated playback rate per plex, and may request or poll monitoring agent 440 for number of maximal outstanding I/O for each disk drive and/or sub-disk in a given desired time window, *e.g.*, every 10 seconds, every 2 minutes, *etc.* Upon receipt of such a request or poll from workload monitor 426, monitoring agent 440 may respond with the requested information, *e.g.*, by sending a disk/subdisk identifier and a respective maximal outstanding I/O for each disk drive and/or sub-disk to the requesting workload monitor 426 of resource manager 420. Monitoring agent 440 may then reset the outstanding I/O counter for the next polling window and start tracking the new value for the next window. For each window, workload monitor 426 may use the number of outstanding I/O per-disk in each plex to estimate the weight of workload distribution into each disk drive and then break down the total number of viewers and the aggregated playback rate on the plex level into the disk level based on the estimated weight of workload distribution per disk drive.

It will be understood that workload monitor 426 and monitoring agent 440 may be implemented in a storage management processing engine using any hardware and/or logical configuration that is capable of performing the described functionalities of same, *e.g.*, the set of tasks specified by monitoring agent 440 to be implemented by logical volume manager 430, and the set of tasks specified by workload monitor 426 to be implemented by resource manager 420. In one exemplary embodiment, monitoring agent 440 and workload monitor 426 may share the same processor space and monitoring agent 440 may keep track of the required information and store it in a table accessible by workload monitor 426 on an as-needed basis.

In either of the above two scenarios, workload monitor 426 of FIG. 4A may report the maximal total viewers per disk drive (MaxNoV_perDisk), and the maximal aggregated playback

rate per disk drive (MaxAggRate_perDisk) to I/O admission controller 424. I/O admission controller 424 may employ a resource model equation or algorithm such as previously described herein. In one exemplary embodiment, I/O admission controller 424 may employ Resource Model Equation (19):

$$\begin{aligned} & \text{MaxNoV_perDisk} * AA / [1 - \text{Reserved_Factor} - \text{MaxAggRate_perDisk} / TR] \leq T \\ & \leq (1 - \text{Reserved_Factor}) * B_{\text{max}} / \{ \text{Buffer_Multiplicity} * [(1 - B_Save) * (\sum_{i=1}^{\text{Nov}} P_i)] \} \end{aligned} \quad (19)$$

In a further exemplary embodiment, workload monitor 426 may be employed to track the following system I/O performance characteristics for each logical volume, for each plex within a logical volume, and for each disk drive within a plex: (1) total number of viewers (“TotalNov”) on a resource (a logical volume, or a plex, or a disk drive), (2) aggregated playback rate (“TotalRate”) on a resource (a logical volume, or a plex, or a disk drive), (3) current weight (“CurrentWeight”) of workload on a disk drive in a plex, and (4) weight of workload on a disk drive (“NewWeight”) based on the latest poll of the outstanding I/O for each disk drive. In this embodiment, Current Weight is initialized to zero and is continuously updated during the course of monitoring. In this embodiment, a configurable parameter α (“Aging_Factor”) may also be employed to update CurrentWeight after a value of NewWeight is obtained from the latest polling window. If desired, the parameter Aging_Factor may be set to a default value, e.g. from about 0.6 to about 0.7.

In this exemplary embodiment, workload monitor 426 may track TotalNov and TotalRate per logical volume, for example, in a manner as previously described. Workload monitor 426 may obtain an estimation of TotalNov and TotalRate per plex by considering the number of plex (“number_of_plex”), for example, as follows:

$$\text{TotalNov_perPlex} = \text{TotalNov_perLV} / \text{number_of_plex} \quad (21)$$

$$\text{TotalRate_perPlex} = \text{TotalRate_perLV} / \text{number_of_plex} \quad (22)$$

Workload monitor 426 may obtain an estimation of TotalNov and TotalRate per disk drive by polling monitoring agent 440 in a manner as previously to find out how the workload is distributed at the disk drive level. Workload monitor 426 may also poll (or access) monitoring agent 440 at the end of each polling window to obtain the maximal outstanding I/O per subdisk in each plex, denoted by “QueueDepth(i)” where i stands for subdisk ID. Workload monitor 426 may then calculate “NewWeight” per subdisk as follows:

$$\text{NewWeight}(i) = \text{QueueDepth}(i) / (\text{Summation of QueueDepth}(j) \text{ for all subdisk } j \text{ in the plex}) \quad (23)$$

To help overcome transient effects and stabilize value of workload weight, the workload monitor 426 may then use the following formula and Aging_Factor α (e.g., default value of about 0.6 to about 0.7) to update “CurrentWeight” per disk drive as follows:

$$\text{CurrentWeight}(i) = \alpha * \text{CurrentWeight}(i) + (1 - \alpha) * \text{NewWeight}(i) \quad (24)$$

Workload monitor 426 may use the following formulas to calculate total number of viewers per subdisk, and total rate per subdisk:

$$\text{TotalNov_subdisk}(i) = \text{TotalNov_perPlex} * \text{CurrentWeight}(i) \quad (25)$$

$$\text{TotalRate_subdisk}(i) = \text{TotalRate_perPlex} * \text{CurrentWeight}(i) \quad (26)$$

Next, workload monitor 426 may aggregate the subdisk level workload (e.g., total number of viewers per subdisk and total rate per subdisk) into a physical disk level workload (e.g., total number of viewers per disk “TotalNov_disk”, and total rate per disk “TotalRate_disk”):

$$\text{TotalNov_disk}(i) = \text{Sum of TotalNov_subdisk}(j) \quad \text{for all subdisk } j \text{ in disk } i \quad (27)$$

$$\text{TotalRate_disk}(i) = \text{Sum of TotalRate_subdisk}(j) \quad \text{for all subdisk } j \text{ in disk } i \quad (28)$$